Adaptive Unsupervised Learning of Human Actions

Arnold Wiliem, Vamsi Madasu, Wageeh Boles, and Prasad Yarlagadda

Queensland University of Technology, Australia, a.wiliem@student.qut.edu.au

Keywords: Anomaly Detection, Security, Video Surveillance System, Computer Vision

Abstract

Automatic detection of suspicious activities in CCTV camera feeds is crucial to the success of video surveillance systems. Such a capability can help transform the dumb CCTV cameras into smart surveillance tools for fighting crime and terror. Learning and classification of basic human actions is a precursor to detecting suspicious activities. Most of the current approaches rely on a non-realistic assumption that a complete dataset of normal human actions is available.

This paper presents a different approach to deal with the problem of understanding human actions in video when no prior information is available. This is achieved by working with an incomplete dataset of basic actions which are continuously updated. Initially, all video segments are represented by Bags-Of-Words (BOW) method using only Term Frequency-Inverse Document Frequency (TF-IDF) features. Then, a data-stream clustering algorithm is applied for updating the system's knowledge from the incoming video feeds. Finally, all the actions are classified into different sets. Experiments and comparisons are conducted on the well known Weizmann and KTH datasets to show the efficacy of the proposed approach.

1 Introduction

Advancements in the computer vision domain have had a direct impact on CCTV surveillance based technologies. Many new methods have been implemented in this domain to carry out some automatic anomaly detections. These methods have made it possible for security professionals to watch many CCTV cameras simultaneously. Several technical challenges need to be addressed to enable effective and efficient detections. In the case where the person's limbs are observable by the systems, most of the current approaches rely on normal human action patterns datasets to detect anomalies [5, 18, 12, 8]. Although this method works well in a constrained environment where all normal human action patterns can be enumerated, in real life scenarios it is almost impossible to get a dataset containing all normal human action patterns (normal datasets) [5]. Another method is to describe normal and/or anomalous actions in the form of rules & constraints [10]. But, this method also has the same drawback.

An effective surveillance system is one which is able to deal with the incomplete normal dataset and/or rules & constraints. The normal dataset and rules & constraints could be used as the initial information. During the operational time, the system gets new information from the video feeds and adds it into its knowledge base. Since the system's knowledge is updated over time, then it would be able to cope better with any unseen actions patterns. For instance, the system may tag an action as anomalous because it is not in the normal dataset. However, when the action appears more frequently, then eventually, it is tagged as normal behaviour. Another example is, an action which previously was tagged as normal which now could be tagged as an anomaly.

One of the methods to make the system able to deal with the incomplete normal dataset is to attempt to discover action patterns continuously. Basically, the process of discovering human action patterns is similar to the data clustering process. It starts by putting similar human actions into one group. Each group is then regarded as a human action pattern. As this process will be done continuously, the system needs to be able to update its knowledge incrementally. In other words, it is assumed that the information is only partially observable by the system as it will do action patterns discovery process before it sees all the human actions.

Currently, the problem of discovering action patterns has received less attention compared to recognition of actions and activities [15]. Current approaches in this domain use techniques assuming that all the information is available. In other words, the system needs to wait till it sees all the human actions before it does the human action patterns discovery process. This assumption does not hold in real life scenarios since discovering action patterns continuously assumes the information to be partially available. The following are some methods and approaches aiming to discover human action patterns.

To discover action patterns, one may contruct action spaces from the extracted features. Liang and Suter [7] contruct action space within a low dimensional manifold embedded in a high dimension. Locality Preserving Projection (LPP) is used for dimensionality reduction. Clustering algorithms can be employed to extract action classes by clustering similar actions [17]. One of the difficulties of employing methods in this line of thinking is the dimensionality reduction method that needs a complete set of action patterns in order to work correctly.

Bag-of-words (BOW) is another way to represent each segment of video as a document. The visual words, in each document are drawn from a corpus quantised spatial motion interest points [5, 3, 11, 13]. Zhong et al [5] use a co-clustering algorithm to cluster both the visual words and the documents. Niebles et al transform the problem into a document retrieval problem and use probabilistic Latent Semantic Analysis (pLSA). This model automatically learns the probability distributions of the spatial-temporal words and the intermediate topics corresponding to human action categories. Inspite of their excellent results shown in experiments, yet, most of them assume that the training sets provide complete action patterns.

Most of the above mentioned approaches only work correctly when they are given a complete set of data representing all possible action patterns within a surveillance scenario. Addressing the posed problem becomes very important when we would like to detect suspicious/unusual action patterns in smart surveillance systems. This is because suspicious action patterns occur so rare that it may not be included in the system's

knowledge. Furthermore, in many cases an action pattern could be tagged as unusual, but later on when its instances become common, it is tagged as normal, and vice versa. Again, this normal action pattern may not be included in the system's knowledge. In order to handle these problems, the system needs continuously maintain its knowledge which contains human action patterns. This work will be the stepping stone into that direction as it describes an approach which is able to continuously update the system's knowledge from the incoming actions.

The proposed approach is based on our previous work which focuses on human trajectory patterns [16]. Since in this work we are dealing with cases where human' limb movements are observable, we are using interest point features. Each video segment is represented as a visual document where the visual words are constructed by clustering the interest points. Technically, Term Frequency-Inverse Document Frequency (TF-IDF) features are used to describe a visual document (i.e. a video segment). We use a modified normalised cosines distance to measure disimilarities between two actions. A data stream algorithm is employed to continuously update the system's knowledge. To our best knowledge, data stream clustering algorithms have never been used to address the posed problem. The experiments are divided into two parts. The first part is a comparative analysis on the features' discriminative power. Even though the focus of our approach is less on the actions classification (i.e. our contribution is on introducing a method to continuously update the system knowledge's by using data-stream algorithm), having a good discriminative features is important. This is because the system won't be able to detect suspicious action patterns when it is not able to distinguish the human actions correctly . The second part is to validate whether or not the approach is able to update the system's knowledge continuously. Two popular publicly available human action datasets were used in the experiments: Weizmann dataset [4] and KTH dataset [14].

2 Feature representation using space-time interest points

Interest points are the local spatio-temporal features in a video segment which are considered salient [2]. These are detected by applying a response function to the video segment in both the temporal and spatial domain. An interest point is detected when there is a region producing high response value. Furthermore, the interest point features are formed by constructing cuboids which include the interest points and their surrounding neighborhoods. These features usually are able to describe the action captured in a video segment. For example, the patches showing a knee bending, a foot is taking off from the ground and a foot landing on the ground could be found in a video segment containing a person jumping.

There are many interest points detectors proposed [14, 2, 3]. Amongst these, perhaps the one proposed by Dollar et al [3] is the most popular one. This is because, it generates a high number of detections which is what the BOW method needs [11].

Technically, interest points are the patches that have undergone a complex motion. In Dollar et al [3], these patches are detected by using Gaussian filters and Gabor filters applied along with temporal axis. Equation 1 defines the response function.

$$R = (I * g * h_{ev})^2 + (I * g * h_{od})^2 \quad (1)$$

where I is a video segment in the form of a cube constructed by stacking up the image sequence; $g(x, y; \sigma)$ is the 2D Gaussian smoothing kernel to be applied in the spatial domain, h_{ev} and h_{od} are the even and odd Gabor filters. These filters are applied on the temporal domain. They are defined as follows.

$$h_{ev}(t; \tau, \omega) = -cos(2\pi t\omega)e^{\frac{-t^2}{\tau^2}} \quad (2)$$

$$h_{od}(t; \tau, \omega) = -sin(2\pi t\omega)e^{\frac{-t^2}{\tau^2}} \quad (3)$$

In their work, ω is defined as $\frac{4}{\tau}$. This means, there are only two free parameters (i.e. τ and σ) which govern the detector scales in the spatial and temporal domains respectively. For simplicity, we follow Niebles et al [11] which use only one scale and rely on the visual words to encode few changes in scale that are observed in the dataset.

As aforementioned, the response function induces a strong impulse at any region undergoing a complex motion. But, any pure translational motion, or without spatially distinguishing features will not induce a strong response [3]. Interest point patches are constructed by including the neighboring pixels in the spatial and temporal domains at a size approximately six times the scales along each dimension.

Despite using the same interest point response function, our method is different from that proposed by Dollar et al [3] and Niebles et al [11] in several ways. Dollar et al use a histogram of visual words frequency of occurrence in a video segment. To measure the dissimilarities between two segments, they use the χ^2 distance function. According to our observation on the affinity matrices constructed on each dataset, there is still some confusion between two different actions (e.g. in Weizmann dataset, bending actions have small distance to the other bending actions and the hand waving actions). This confusion affects the system performance as can be seen in the experiment results. Furthermore, Niebles et al introduce the use of probabilistic latent semantic analysis (pLSA) which assumes that there is latent topic (i.e. action class) in each video segment. In spite of the excellent performance, their method is difficult to extend to environments where the number of action classes is unknown.

In our method, we use TF-IDF feature which are one of features the most commonly used in information retrieval domain [9]. This features were later applied by Jingen et al [6] to recognise human action captured in a video feed. In their work, they fuse these features with the other silhouette based features. They successfully showed that by fusing these two features, a better system performance could be achieved. In our case, we decided to use the TF-IDF features alone because it does not need any low level vision processes (e.g. background subtraction, tracking, etc) which are necessary in the case of any silhouette based feature.

In the document retrieval domain, TF-IDF features are used as vector representation of a document. TF-IDF feature is formed by multiplying both TF and IDF features. Each TF (Terms Frequency) feature contains the frequency of word found in a document. TF features can be defined as follows. Let $W = \{w_1, w_2, ..., w_n\}$ be the set of words found in documents. The TF vector of a document is an n-dimensional vector (n equals to the number of words in W) which its i-th entry contains the frequency of word w_i found in that document. IDF (Inverse Document Frequency) features are derrived from Document Frequency (DF) features by using equation 4. DF (Document Frequency) feature is similar to TF feature. The only difference is that its i-th entry contains the number of documents which the word w_i are found.

In our case, each video segment/clip is regarded as a document. As shown in Figure 1, the set of words W is constructed by employing K-means clustering on the extracted patches of interest points. Each cluster which is represented by its centre

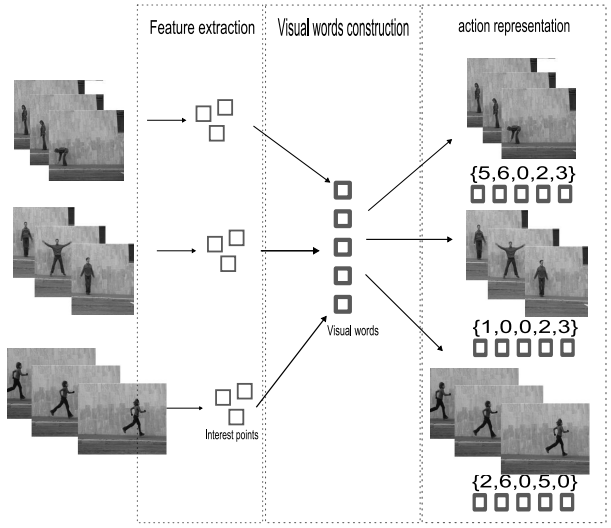


Figure 1: Feature extraction process. The first step is that interest point patches are extracted from each video segment. These patches are used to form the set of visual words by clustering them into k clusters. Once the set of visual words is constructed, each video segment will be described by using TF-IDF features

(cluster mean), is regarded as a word. These words are regarded as visual words.

$$idf_t = log(\frac{N}{df_t}) \tag{4}$$

where idf_t is the IDF of the t-th word; N is the number of documents;df_t is the number of documents in which the words occurs.

One of advantages of using TF-IDF features over TF only features [9] is that IDF features give high weight for any words that are found within a small number of documents. This would lead to high discriminating power to those documents. In other words, low IDF value means that the word appears in many documents. Furthermore, the word having the lowest IDF value virtually appears in every document.

Finally, a normalised cosines distance function is applied to measure the dissimilarities between two video segments. The equation below presents the normalised cosines distance function.

$$d(x_i, x_j) = \frac{1}{e^{(\frac{x_i \cdot x_j}{|x_i||x_j|})}} \tag{5}$$

where x_i and x_j are the feature vectors of two different video segments. The distance function has range $[\frac{1}{e}, e]$. The lowest and highest value occur when the angle between both vectors is 0, and π respectively.

3 Data Stream Clustering Algorithm

A data stream is a data model where each of its data points can only be accessed sequentially in an ordered manner. Mathematically, a data stream can be defined as an ordered sequence of points $x_1, ..., x_n$ where n could be unbounded (i.e. $n \approx \infty$) [16]. The followings are the properties of a data stream model: (1) The data elements arrives online. (2) The system cannot choose the order of which data will be read. In other words, the system does not have any control on ordering of the data . (3) As mentioned before, it is potentially unbounded. (4) Generally, once an element from a data stream has been processed, it has to be discarded or archived. This is because the size of the the data stream that is unbounded compared to the system's memory.

As shown in our previous work [16], surveillance data can be modeled as a large scale data stream model. This also is one of the reasons why one will never have sufficient normal datasets, because, by assuming it as a data stream model, it means that we only have partial knowledge about all the data. The system's knowledge needs to be updated as the new information arrives. So, there is a chance that there is a normal action which has not been accounted for yet in the system's knowledge. This normal action could appear in the future, and the system has to be able to learn eventually that this action is normal. A data stream clustering algorithm can be used to address this problem. In this work, we modify the data stream clustering algorithm proposed in our previous work [16]. We use multiple points to represent the cluster centroid instead of using a single point, and we use these multiple points to define distance between a data point to a cluster. Since, a cluster may not have sufficient information to decide its cluster centroid, using multiple representations gives a better approximation of its cluster centroid.

By using this data stream clustering algorithm, the system will be able to cluster any type of action not existing in the current system's knowledge. Hence, it could operate under circumstances where it is impossible to enumerate all kinds of actions. Furthermore, in order to cope with the unbounded data size, the pyramidal time frame concept proposed by Aggarwal et al [1] could be used. Principally, pyramidal time frame maintains a hashtable containing snapshots of the clustering result saved at a particular time. These snapshots provides a sufficient summary of the previous data which may have been archived or deleted. Therefore, the system is able to discover action patterns at any given time frame. Before using this concept, one needs to have a better understanding about the performance of the algorithm when all the data can be fitted in the system. Hence, in this precursor study, we only show the algorithm performance when all the data can be handled within the system's memory.

The following are the clustering algorithm descriptions in brief. A more detailed explanation is available in [16]. Let $\dot{x}$ be an instance of a data point which represents each feature vector. Let W be the set of clusters. Let C be the set of clusters and c_i be a cluster index i. So, $C = \{c_1, c_2, c_3, ..., c_n\}$. Let $\dot{c_i}$ be the cluster centroid of cluster c_i. Since a cluster may have multiple points representing its centroid then we define $\dot{c_i} = \{\dot{c_{i1}}, ..., \dot{c_{in}}\}$, where n is the maximum representations number of which a cluster can have. A cluster has one representation which represents its centroid if and only if it is a singleton cluster (i.e. a cluster having only one member). Each cluster centroid is not derived by taking the mean of cluster members. Each representation is chosen as one of the cluster members which has the t-th smallest distance to the other members. By using it, the algorithm only needs affinity/distance matrix information. Equation 6 depicts the formula to get a cluster t-th representation.

$$\dot{c_{it}} = \arg\min_j (\sum_{x_j, x_k \in c_i} dist(x_j, x_k)), \; where \, i \neq j \tag{6}$$

We define distance function as a function which calculates the dissimilarity between two features. In this case we use the normalised cosines distance function. Let Lv be the threshold that defines the smallest cluster. Lv decides whether an instance of a data point should be clustered into a new cluster or

Figure 2: Some feeds taken from both datasets. The first and last two rows are taken from Weizmann dataset and KTH dataset respectively.

be merged into one of the existing ones. In addition, Lv also decides whether two clusters should be merged or not. Initial clustering is performed on training sets to construct the initial behaviour clusters structure and to calculate Lv. This initial clustering can use any suitable off-line/on-line clustering algorithm. The initial datasets may not contain all possible action patterns. The algorithm will update its knowledge once it observes unseen action patterns. Equation 7 shows how Lv is derived from training sets.

$$L_v = \frac{1}{N} \sum_{c_i \in C} (Lv_{c_i}) \tag{7}$$

where Lv_{c_i} is the average distance from each member of the cluster c_i to its first class representation $\dot{c_{i1}}$, and N is the current number of clusters. Equation 8 shows the equation for Lv_{c_i}

The distance between a data point and a cluster is calculated using equation 9. This will handle the case where the cluster is having multiple points representing its centroid.

$$Lv_{c_i} = \frac{1}{n_{c_i} - 1} \sum_{x_j \in c_i} dist(\dot{c_{i1}}, x_j) \tag{8}$$

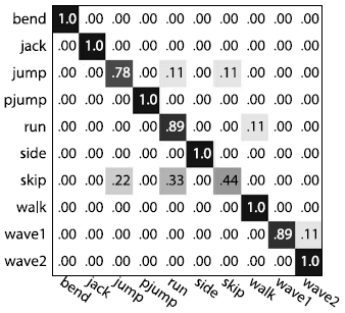
where n_{c_i} is the number of cluster c_i members. Lv_{c_i} is updated every time a new member is added into the cluster.

$$dist(x, \dot{c_i}) = \begin{cases} dist(x, \dot{c_{i1}}) \; if \; c_i \; singleton \\ mean_t(dist(x, \dot{c_{it}})) \; otherwise \end{cases} \tag{9}$$

In general, the clustering algorithm can be described in the following steps: (1) Construct an initial dataset which can be gathered from the actions observed by the system so far. (2) Perform an initial clustering using any online/offline clustering algorithm. The algorithm may need the number of cluster as one of the required parameters. Since, the clustering is performed on the manually-constructed initial dataset, then one could use the prior knowledge, or some statistical methods such as the gap statistic [19]. (3) Calculate the spread threshold value from the initial clustering C, and wait until a new data point is observed. (4) For any observed data point x, find the closest cluster to x. (5) If the distance between x and the cluster is not statistically large (i.e. twice as large) from Lv or L_{Vc} (the algorithm uses Lv when the cluster is a singleton cluster, or L_{vc} otherwise) then: put x into the cluster, update the cluster properties and check whether the cluster needs to be merged to another cluster. Two clusters are merged when the distance between the clusters' centroids is not statistically large from Lv. (6) Otherwise, create a new singleton cluster, and put x into it.

4 Experiments and Discussion

We have divided the experiment into two sections. First, comparative experiments will be presented to show the performance of the features discriminative power. The experiments showing the performance of the clustering algorithm in addressing the presented problem will be shown next.



(a)

Figure 3: Result of recognition test on Weizmann dataset taken from Niebles et al work [11].

We applied our approach on two publicly available datasets: Weizmann dataset [4] and KTH dataset [14].

Weizmann dataset contains 90 video clips from 9 different subjects. Each video clip contains one subject performing a single action. There are 10 different action categories: bending, jumping, jumping jack, jumping in a place, running, sideways, skipping, walking, one-hand waving and two-hand waving. Each clips lasts about 2 seconds at 25 Hz with image frame size of 180x144. Figure 2 shows some of the examples taken from the dataset.

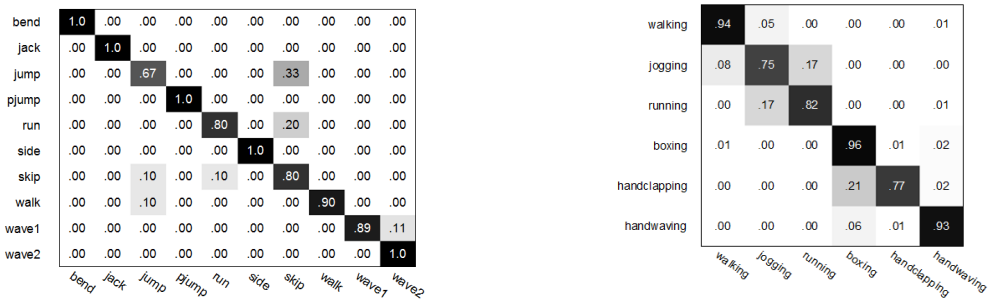
KTH dataset is the largest public human activity video dataset. It has 6 kinds of actions: boxing, hand clapping, hand waving, jogging, running and walking. There are 25 different subjects done these 6 actions. In addition it has some variations: indoor, outdoor, changes in clothing and variations in scale. Each video segment contains only one subject performing a single action. Each subject has around 23 to 24 segments. In total, there are 599 video segments. Each video segment is sampled at 25Hz and lasts between 10 to 15 seconds with image frame size of 160x120. Figure 2 shows some of the examples taken from the dataset.

4.1 Parameters

As mentioned in the previous sections, the set of visual words is one of the important components in this approach because it is used for describing a video segment. There are a number of parameters for constructing the visual words: the parameters determining the interest points spatio-temporal scale (i.e. τ and σ), the number of visual words, and the number of video segments used to construct visual words.

We set τ and σ differently on each dataset. This is because, both have different spatio-temporal scales. We set $\tau = 2.5$ and $\sigma = 2$ on KTH dataset. Both τ and σ are set to 1.2 on Weizmann dataset. We derived these values based on [11] and some observations confirming the suitability of those values.

Based on our observations, 1200 visual words produce a reasonable performance on both datasets. In addition, we found that although setting the correct number of visual words is important, what is more important is the completeness of the visual words. Some words are only found in a particular action class. If these words are missing, the discriminative power of the features will decrease. In this work, this issue is addressed by including video segments on each action class when constructing the visual words. Another issue is determining the number of video segments needed to construct such visual words. The choice is made based on the recognition rate performance on each dataset.



(a) (b)

Figure 4: The best recognition test results. (a) Results from Weizmann dataset, where 70 video clips were used to construct visual words; (b) Results from KTH dataset, where 120 video clips were used to construct visual words.

4.2 Evaluation

In this first experiment we test the features' discriminative power performance by using Leave-One-Out (LOO) test scenario used by the other approaches. In this scenario each test only consists of one video segment. All actions done by the same subject as the test data are excluded from the training sets. To make the experiment results comparable with the other methods, we use the Nearest Neighbour (K-NN) method with K = 1 as the classifier.

Figure 4 depicts the best recognition result achieved by the features. As we can see, the approach generates less confusion on Weizmann dataset (i.e. better recognition result). This is because Weizmann dataset has less variations. Despite its better performance, there are still big confusions between actions jump and skip. The reason why this happens is that these actions are described using similar visual words. As we can see from figure 3, this also happens in Niebles et al's approach [11] which uses BOW as its base. This indicates that there are other aspects needed to be exploited in order to describe some actions (e.g. the order of visual words [2]).

Table 1 depicts the best recognition results from each dataset and the comparisons with other BOW methods. Generally the proposed approach has a better recognition rate compared with the other basic BOW methods. This means that the features have a better discriminative power. Having features with good discriminative power will help the data-stream clustering algorithm to cluster incoming data correctly.

Actually, there are other BOW methods [2, 20] having a better recognition rate, nevertheless they are not suitable to address the presented problem. In [2], it uses a global optimisation method which is not feasible to be applied in our assumptions where the algorithm only partially sees the data. Furthermore, the method in [20] uses pLSA which needs all the data be available to do Locality Preserving Projection (LPP) dimensionality reduction. This issue also happens in Niebles et al method [11].

The next experiment is to test whether the proposed approach is good enough when dealing with unseen patterns. This means, given the initial sets, the method should be able to put instances of unseen action patterns from the same class into a new cluster. To test this, we set up a scenario similar to the LOO scenario with the only difference being that we put all actions patterns from the same class as the one being tested into the stream part. In other words, the system will be given an initial dataset containing all action patterns except the test instance and all action patterns from the same class as the test instance. Then, the action patterns from the same class as the test instance will be supplied to the algorithm in a stream manner. Finally, the test instance will be supplied to the system.

Table 1: Comparative results with other BOW methods on the KTH and Weizmann datasets.

Method	KTH	Weizmann
Proposed approach	**86.097 %**	**90.22 %**
Niebles et al [11]	83.3 %	90.0 %
Dollar et al [3]	81.17 %	85.2 %

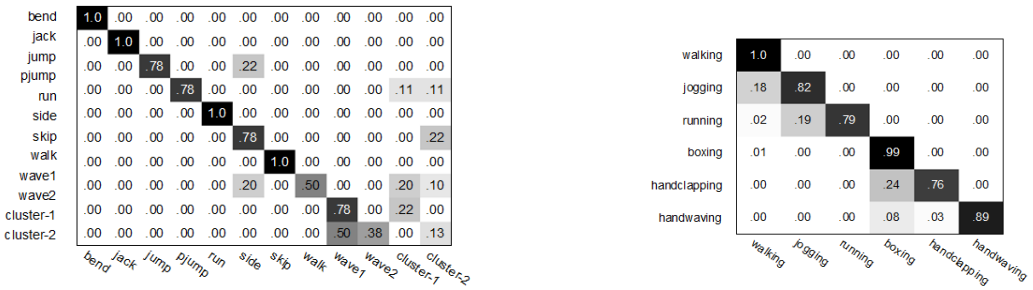
So, there are three possibilities for how a test instance will be classified: it will be classified into a new non-singleton cluster, another existing cluster and a new singleton cluster. We only label the first possibility as the correct result.

The results from this experiment are depicted in figure 5. 76.08% of instances in Weimann dataset were correctly classified. The result from KTH dataset is 84.22% which is better than the Weizmann dataset. As we can see in Weizmann dataset, there are some test instances clustered into a singleton cluster. However, it does not happen in KTH dataset. This phenomenon is related to algorithm convergence. As noted in [16], the algorithm will divide a class into several clusters when there is not enough number of instances belonging to the class. However, in KTH dataset, we observed that the algorithm merged some overclustered clusters into one because there are more instances in the dataset. This explains why we don't see any test clustered into a new singleton cluster.

Note that there is performance loss on the second experiment results compared with the first experiment results. This is due to the features discriminative power and the data-stream clustering algorithm convergence rate. As mentioned in the second experiment modified LOO scenario the only way a test instance is labeled as correct is when it is clustered into a new non-singleton cluster. This means that to correctly classify the test instance, the system relies not only on the distances between the instance and the existing classes but also the L_v and Lv_{ci} which rely on the features discriminative power. The ideal case is where the features discriminate perfectly between two different features coming from different classes. Previously, it has been discussed that the data-stream algorithm divides a newly observed pattern class into several clusters. These clusters will be merged together when there is a large enough number of instances. This also explains why Weizmann dataset having less number of instances, has a bigger decrement compared with the KTH dataset in the second experiment.

5 Conclusions

Recent advancements in video surveillance technologies have had a significant impact on the security domain. One of the focus area is suspicious activity detection which would help se-



(a) (b)

Figure 5: Results showing the robustness of the method when dealing with unseen patterns (a) Results from Weizmann dataset. Number of points representing a cluster centroid: 10; (b) Results from KTH dataset. Number of points representing a cluster centroid: 50

curity officers to monitor many surveillance areas at any given time. However, there are several issues and challenges that need to be addressed before such an automatic detection system can be implemented. Many of the current methods proposed withing this field rely on the assumption that there is a dataset defining a complete set of normal human action patterns.

In this paper, we presented an approach which is able to deal with unseen patterns by classifying them into different classes. This is of immense significance to a video surveillance system which can use its basic action patterns dataset as initial information and update this knowledge base continuously with each incoming video feed. To achieve this, we proposed a BOW method using only TF-IDF features for describing a video segment. The distance between two video segments is calculated via normalised cosine distance function. In addition, a data-stream method for detecting anomalies is employed to make the system deal with unseen patterns. Experiments showed that the proposed method achieved better performance as compared to other basic BOW methods on both Weizmann and KTH datasets. It was also shown that this approach is better suited for real life scenario as suspicious action patterns can be now be detected continuously in real time. However, more work needs to be done to perfect the system. One of the key issues is to devise a way to guarantee the completeness of visual words.

References

[1] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, "A framework for clustering evolving data streams," in *the 29th VLDB Conference*, Berlin, Germany, 2003, pp. 81–92.

[2] M. Bregonzio, S. Gong, and T. Xiang, "Recognising action as clouds of space-time interest points," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, Miami, USA, 2009.

[3] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*, 2005, pp. 65–72.

[4] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 12, pp. 2247–2253, 2007.

[5] Z. Hua, S. Jianbo, and M. Visontai, "Detecting unusual activity in video," in *Computer Vision and Pattern Recognition. CVPR. Proceedings of the IEEE Computer Society Conference on*, vol. 2, 2004, pp. 819–826.

[6] L. Jingen, S. Ali, and M. Shah, "Recognizing human actions using multiple features," in *Computer Vision and Pattern Recognition. CVPR. IEEE Conference on*, 2008, pp. 1–8.

[7] W. Liang and D. Suter, "Learning and matching of dynamic shape manifolds for human action recognition," *Image Processing, IEEE Transactions on*, vol. 16, no. 6, pp. 1646–1661, 2007.

[8] D. Makris and T. Ellis, "Learning semantic scene models from observing activity in visual surveillance," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 35, no. 3, pp. 397–408, 2005.

[9] C. D. Manning, P. Raghavan, and H. Schutze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[10] R. Martnez-Toms, M. Rincn, M. Bachiller, and J. Mira, "On the correspondence between objects and events for the diagnosis of situations in visual surveillance tasks," *Pattern Recognition Letters*, vol. 29, no. 8, pp. 1117–1135, 2008.

[11] J. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised learning of human action categories using spatial-temporal words," *International Journal of Computer Vision*, vol. 79, no. 3, pp. 299–318, 2008.

[12] C. Piciarelli and G. L. Foresti, "On-line trajectory clustering for anomalous events detection," *Pattern Recognition Letters*, vol. 27, no. 15, pp. 1835–1842, 2006.

[13] S. Savarese, A. DelPozo, J. C. Niebles, and F.-F. Li, "Spatial-temporal correlatons for unsupervised action classification," in *Motion and video Computing. WMVC. IEEE Workshop on*, 2008, pp. 1–8.

[14] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local svm approach," in *Pattern Recognition. ICPR. Proceedings of the 17th International Conference on*, vol. 3, 2004, pp. 32–36 Vol.3.

[15] P. Turaga, A. Veeraraghavan, and R. Chellappa, "Unsupervised view and rate invariant clustering of video sequences," *Computer Vision and Image Understanding*, vol. 113, no. 3, pp. 353–371, 2009.

[16] A. Wiliem, V. Madasu, W. Boles, and P. Yarladgadda, "A context-based approach for detecting suspicious behaviours," in *Proceedings of the Digital Image Computing: Techniques and Applications*, 2009.

[17] W. Xiaozhe, W. Liang, and A. Wirth, "Pattern discovery in motion time series via structure-based spectral clustering," in *Computer Vision and Pattern Recognition. CVPR. IEEE Conference on*, 2008, pp. 1–8.

[18] Z. Yue, Z. Yue, Y. Shuicheng, and T. S. Huang, "Detecting anomaly in videos from trajectory similarity analysis," in *Multimedia and Expo, IEEE International Conference on*, Y. Shuicheng, Ed., 2007, pp. 1087–1090.

[19] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Advances in Neural Information Processing System*, vol. 17. MIT Press, 2004, pp. 1601–1608.

[20] Z. Zhang, Y. Hu, S. Chan, and L.-T. Chia, "Motion context: A new representation for human action recognition," in *Computer Vision ECCV*, 2008, pp. 817–829.